Claims

[c1] What is claimed is: 1. In an object-oriented computer system, a method of generating software components, the method comprising the steps of:
a) Loading a structure of an object to a memory means;
b) Loading the file structure data by a file component loader; and
c) Interafacing between the structure of an object and the file structure data by an interface object.

[c2] 2. The method of claim 1 wherein step c further comprises: c) interfacing between the structure of an object and the file structure data by an interface object using the methods InitComplexRepository(), GetComplexData() and PutComplexData(). 3. The method of claim 1 wherein step a further comprises: a) said structure contains a plurality of pointers to complex functionalities and complex data storage buffers. 4. The method of claim 1 wherein step b further comprises:b) Loading the file structure data by a file component loader which includes complex functionality and a knowledge base,5. The method of claim 1 wherein step c further comprises: c) interfacing between the structure of an object and the file structure data by an interface object consisting of the methods of linking memory references, pointers and buffers, geting the proper pointers to the data in the repository ofr loading into the structure of an object, retrieving data from a repository to be used by the structure, writing data to a repository, and releasing all memory references, pointers and buffers that belongs to a repository link.

[c3] 6. The method of claim 1 which includes step d comprising of: d. Maintaining a knowleged base that contains the acquired knowledge data of the component operation. 7. The method of claim 1 wherein step a further comprises:a) said structure contains a plurality of pointers to complex functionalities and complex data storage buffers and said complex data storage buffers includes metadata models.

[c4] 8. The method of claim 1 wherein step a further comprises:
a) said structure contains a plurality of pointers to complex functionalities and complex data storage buffers and said complex data storage buffers includes

pattern recognization components.

[c5] 9. In an object-oriented computer system, a method of generating software components, the method comprising the steps of:a. Allocating an object in a memory means;

b. Copying the structure to a memory means;

c.Placing the records of complex data, complex functionality and knowledge base in the Complex Data Storage Buffer;

d.Creating complex data at instantiation moment;

e.Modifying complex data;

f.Deciding whether to write complex data, complex functionality and knowledge base to a memory means;

g.Exposing the structures and functionality through an interface that performs all specific task of the object itself and administrative tasks related to the structure; and

h.Using a object in which all structures can be used like memory structures and all memory allocations and swapping will be managed by the object itself.

[c6] 10. In an object-oriented computer system, a device to generate software components comprising of:

a) a load means for loading a structure of an object in a memory means;

b) a load means for loading the file structure data by a file component loader; and

c) an interface means for interafacing between the structure of an object and the file structure data by an interface object.

[c7] 11. The device of claim 10 in which: c) the interface means between the structure of an object and the file structure data by an interface object using the methods InitComplexRepository(), GetComplexData() and PutComplexData(). 12. The device of claim 10 in which:a) said structure contains a plurality of pointers to complex functionalities and complex data storage buffers. 13. The device of claim 10 in which:b) the load means loads the file structure data which includes complex functionality and a knowledge base by a file component loader,14. The device of claim 10 in which:c) the interface means between the structure of an

object and the file structure data by an interface object using the methods of linking memory references, pointers and buffers, initializing and geting the proper pointers to the data in the repository ofr loading into the structure of an object, retrieving data from a repository to be used by the structure, writing data to a repository, and releasing all memory references, pointers and buffers that belongs to a repository link.

[c8] 15. The device of claim 10 in which:a knowleged base is maintained that contains the acquired knowledge data of the component operation. 16. The device of claim 10 in which:said structure contains a plurality of pointers to complex functionalities and complex data storage buffers and said complex data storage buffers includes metadata models.

[c9] 17. The device of claim 10 in which:
said structure contains a plurality of pointers to complex functionalities and complex data storage buffers and said complex data storage buffers includes pattern recognization components.

[c10] 18. In an object-oriented computer system, a device of generating software components comprising:
a. An allocation means to allocate an object in a memory means;
b. a writing means to copy structure to a memory means;
c.The placing the records of complex data, complex functionality and knowledge base in the Complex Data Storage Buffer;
d.The creating of complex data at instantiation moment;
e.The modification of the complex data;
f.Using a decision means to decide whether to write complex data, complex functionality and knowledge base to a memory means;
g.The exposing of the structures and functionality through an interface that performs all specific task of the object itself and administrative tasks related to the structure; and
h.The Using of a object in which all structures can be used like memory structures and all memory allocations and swapping will be managed by the object itself.

[c11] 19. A computer program product for generating software components, the computer program product comprising a computer usable medium having computer readable program code thereon, including: program code for Loading a structure of an object in a memory means;
program code for loading the file structure data by a file component loader; and program code for Interfacing between the structure of an object and the file structure data by an interface object.

[c12] 20. The computer program product of claim 19 wherein the base component has interfaces and the program code for:interfacing between the structure of an object and the file structure data by an interface object using the methods InitComplexRepository(), GetComplexData() and PutComplexData(). 21. The computer program product of claim 19 wherein the base component has interfaces and the program code for:having the structure containing a plurality of pointers to complex functionalities and complex data storage buffers. 22. The computer program product of claim 19 wherein the base component has interfaces and the program code for:program code for loading the pertinent data of the file extension by a file component loader which includes complex functionality and a knowledge base,23. The computer program product of claim 19 wherein the base component has interfaces and the program code for:interfacing between the structure of an object and the pertinent data of the file extention by an interface object consisting of the methods of linking memory references, pointers and buffers, geting the proper pointers to the data in the repository ofr loading into the structure of an object, retrieving data from a repository to be used by the structure, writing data to a repository, and releasing all memory references, pointers and buffers that belongs to a repository link.

[c13] 24. The computer program product of claim 19 wherein the base component has interfaces and the program code for:Maintaining a knowleged base that contains the acquired knowledge data of the component operation. 25. The computer program product of claim 19 wherein the base component has interfaces and the program code for:said structure to contains a plurality of pointers to complex functionalities and complex data storage buffers and said

complex data storage buffers includes metadata models.

[c14] 26. The computer program product of claim 19 wherein the base component has interfaces and the program code for:
said structure to contains plurality of pointers to complex functionalities and complex data storage buffers and said complex data storage buffers includes pattern recognization components.

[c15] 27. The computer program product of claim 19 wherein the base component has interfaces and the program code for:
a. Allocating an object in a memory means;
b. Copying the structure to a memory means;
c.Placing the records of complex data, complex functionality and knowledge base in the Complex Data Storage Buffer;
d.Creating complex data at instantiation moment;
e.Modifying complex data;
f.Deciding whether to write complex data, complex functionality and knowledge base to a memory means;
g.Exposing the structures and functionality through an interface that performs all specific task of the object itself and administrative tasks related to the structure; and
h.Using a object in which all structures can be used like memory structures and all memory allocations and swapping will be managed by the object.